

Amendments to the Claims:

1. (original) An apparatus comprising a processor and a memory storing computer program code, wherein the memory and stored computer program code are configured, with the processor, to cause the apparatus to at least:  
~~computing device comprising~~

implement a scheduler incorporating an algorithm for ordering the running of threads of execution having different priorities, the scheduler maintaining a ready ;~~and~~  
~~including a list of threads which are scheduled to run on the device, ordered by priority;~~  
and

~~the device further comprising~~ implement ~~at least one locking mechanism for~~  
~~blocking configured to block~~ access to a resource of the device ~~from all threads except~~  
for a thread that holds the locking mechanism,

wherein in an instance in which; and in which the scheduler selects a thread on the ready list to run, but the selected thread a scheduled thread which is blocked from running because a resource it requires is blocked, the scheduler does not switch to the blocked thread but retains the blocked thread in its place by priority on the ready list and instead yields to the thread which holds the locking mechanism and causes the thread which holds the locking mechanism to run.

2. (Currently Amended) An apparatus ~~computing device~~ according to claim 1  
wherein states are assigned to threads and the ready list comprises of all threads  
having a common state.

3. (Currently Amended) An apparatus ~~computing device~~ according to claim 2  
wherein a blocked thread is not permitted to change its state.
4. (Currently Amended) An apparatus ~~computing device~~ according to claim 1  
wherein the ready list is subdivided in accordance with the priority of the threads  
it contains.
5. (Currently Amended) An apparatus ~~computing device~~ according to claim 1  
wherein a thread ~~is arranged to~~ contains a pointer to any locking mechanism it is  
blocked on.
6. (Currently Amended) An apparatus ~~computing device~~ according to claim 1  
wherein the memory and stored computer program code are configured, with the  
processor, to cause the apparatus to implement ~~comprising~~ a plurality of non-  
nestable locking mechanisms.
7. (Currently Amended) An apparatus ~~computing device~~ according to claim 1  
wherein the memory and stored computer program code are configured, with the  
processor, to cause the apparatus to call the scheduler ~~is arranged to be called~~ at  
the end of an interrupt service routine which is caused to run ~~on the computing~~  
~~device~~.
8. (Currently Amended) An apparatus ~~computing device~~ according to claim 1  
wherein the locking mechanism(s) comprise(s) a mutex including a pointer, which

is null if the mutex is free or points to the thread holding the mutex, and includes a flag indicating whether or not the mutex is contested.

9. (Currently Amended) An apparatus ~~computing device~~ according to claim 8 wherein the algorithm is ~~arranged~~ configured to delegate memory management to a replaceable memory model configured in dependence upon ~~[[the]]~~ a configuration of the ~~computing device~~ apparatus.
10. (Currently Amended) An apparatus ~~computing device~~ according to claim 9 wherein the memory model is ~~arranged~~ configured to run in either pre-emptible or non-preemptible modes.
11. (Currently Amended) An apparatus ~~computing device~~ according to claim 10 wherein a mutex is ~~arranged~~ configured to protect the module from running in the pre-emptible mode.
12. (Currently Amended) An apparatus ~~computing device~~ according to claim 1 wherein the scheduler is included in a kernel of an operating system of the ~~computing device~~ apparatus.
13. (Currently Amended) An apparatus ~~computing device~~ according to claim 12 wherein the kernel comprises a microkernel or a nanokernel, and wherein the threads are, respectively, microkernel or nanokernel threads.

14. (Currently Amended) An apparatus ~~computing device~~ according to claim 12 wherein the memory and stored computer program code are configured, with the processor, to cause the apparatus to call the scheduler ~~is arranged to be called~~ each time the kernel is unlocked.
15. (Currently Amended) An apparatus ~~computing device~~ according to ~~any~~ claim 1, wherein the apparatus comprises ~~comprising~~ a mobile computing device.
16. (Currently Amended) An apparatus ~~computing device~~ according to claim 15, wherein the mobile computing device comprises ~~comprising~~ a smart phone.
17. (Currently Amended) A method ~~of operating a computing device, the method~~ comprising:  
providing a scheduler incorporating an algorithm for ordering the running of threads of execution having different priorities, the scheduler maintaining a ready and  
~~including~~ a list of threads which are scheduled to run on ~~[[the]]~~ a computing device,  
ordered by priority; and  
providing at least one locking mechanism ~~for blocking~~ configured to block access to a resource of the device from all threads except for a thread that holds the locking mechanism,  
wherein in an instance in which the scheduler selects a thread on the ready list to run, but the selected thread is blocked from running because a resource it requires is

blocked, the scheduler does not switch to the blocked thread but retains the blocked thread in its place by priority on the ready list and instead yields to the thread that holds the locking mechanism and causes the thread that holds the locking mechanism to run.

~~;~~ ~~and arranging for a scheduled thread which is blocked from running to cause the thread which holds the locking mechanism to run.~~

18. (Currently Amended) [[An]] A non-transitory computer-readable storage medium storing computer program code for an operating system for a computing device,  
the operating system comprising:

a scheduler incorporating an algorithm for ordering the running of threads of execution having different priorities, the scheduler configured to maintain a ready and including means for providing a list of threads which are scheduled to run on the device, ordered by priority; and

at least one locking mechanism ~~for blocking~~ configured to block access to a resource of the device from all threads except for a thread that holds the locking mechanism,

wherein in an instance in which the scheduler selects a thread on the ready list to run, but the selected thread is blocked from running because a resource it requires is blocked, the scheduler is configured to not switch to the blocked thread but retain the blocked thread in its place by priority on the ready list and instead yield to the thread that holds the locking mechanism and cause the thread that holds the locking mechanism to run.

~~;~~ and means for arranging for a scheduled thread which is blocked from running because the resource it requires is locked to cause the thread which holds the locking mechanism to run.

19. (Currently Amended) A[[n]] method ~~operating system~~ according to claim [[18]]  
17 wherein states are assigned to threads and the list comprises of all threads  
having a common state.
20. (Currently Amended) A[[n]] method ~~operating system~~ according to claim 19,  
wherein arranged to inhibit a blocked thread is inhibited from changing its state.
21. (Currently Amended) A[[n]] method ~~operating system~~ according to claim [[18]]  
17, further comprising subdividing arranged to subdivide the ready list in  
accordance with the priority of the threads it contains.
22. (Currently Amended) A[[n]] method ~~operating system~~ according to claim [[18]]  
17 wherein a thread ~~is arranged to~~ contains a pointer to any locking mechanism it  
is blocked on.
23. (Currently Amended) A[[n]] method ~~operating system~~ according to claim [[18]]  
17 comprising providing a plurality of non-nestable locking mechanisms.

24. (Currently Amended) A[[n]] method ~~operating system~~ according to claim [[18]]  
17 wherein the scheduler is ~~arranged to be~~ called at the end of an interrupt service  
routine which is caused to run on the computing device.
25. (Currently Amended) A[[n]] method ~~operating system~~ according to claim [[18]]  
17 wherein the locking mechanism(s) comprise(s) a mutex including a pointer,  
which is null if the mutex is free or points to the thread holding the mutex, and  
includes a flag indicating whether or not the mutex is contested.
26. (Currently Amended) A[[n]] method ~~operating system~~ according to claim 25  
wherein the algorithm is ~~arranged~~ configured to delegate memory management to  
a replaceable memory model configured in dependence upon [[the]] a  
configuration of the computing device.
27. (Currently Amended) A[[n]] method ~~operating system~~ according to claim 26  
wherein the memory model is ~~arranged~~ configured to run in either pre-emptible or  
non-preemptible modes.
28. (Currently Amended) A[[n]] method ~~operating system~~ according to claim 27  
wherein a mutex is ~~arranged~~ configured to protect the module from running in the  
pre-emptible mode.

29. (Currently Amended) A[[n]] method ~~operating system~~ according to claim [[18]]  
17 wherein the scheduler is included in [[the]] a kernel.
30. (Currently Amended) A[[n]] method ~~operating system~~ according to claim 29  
wherein the kernel comprises a microkernel or a nanokernel and wherein the  
threads are, respectively, microkernel or nanokernel threads.
31. (Currently Amended) A[[n]] method ~~operating system~~ according to claim 29  
wherein the scheduler is ~~arranged to be called~~ each time the kernel is unlocked.